Appl. No. 10/724,863
Response Dated May 15, 2006
Reply to Office Action of February 15, 2006

Amendments to the Specification

Please add new paragraphs 12 and 13 to the specification as follows:

[0012] **FIG. 10**, illustrates a diagram of a first level store queue comprising an n-entry circular buffer holding the last n stores in the instruction window with head and tail pointers, an address matching circuit, and a store select circuit, according to embodiments of the present inventions.

[0013] **FIG. 11**, illustrates a diagram of a second level circuit comprising a memory dependence predictor and an unresolved address buffer, according to embodiments of the present inventions.

Please amend original paragraphs 16, 17, 27, 28, 29, and 55 as follows:

[0016] In many embodiments of the present invention, the L1 STQ may be utilized as the principle store queue. The L1 STQ may be a small n-entry buffer holding the last n stores in the instruction window. This buffer may be designed as a circular buffer with head and tail pointers. According to some embodiments, when a new store is inserted into the instruction window, an entry may be allocated for the store at the tail of the L1 STQ. The L1 STQ, in order to prevent stalling when it is full, may remove the previous store from the head of the queue to make space for the new store. The previous store may either be moved into a backing L2 CT, according to some embodiments of the present invention, or into a speculative data cache, according to other embodiments of the present invention.

2

Appl. No. 10/724,863
Response Dated May 15, 2006
Reply to Office Action of February 15, 2006

The L1 STQ may have the necessary address matching and store select circuit to forward

data to any dependent loads. (See FIG. 10, for example.)

[0017] With respect to FIG. 1, a diagram of a store queue with a second level circuit is

shown, according to an embodiment of the present invention. In this diagram,

hierarchical store queue 100 is shown to include at least, but not limited to, a L1 STQ 102

coupled to a L2 CT 104. Both elements 102 and 104 are capable of receiving stores

independently. L2 CT 104 is further capable of receiving stores from the L1 STQ 102.

The L1 STQ 102 may include address matching and store select circuit to forward data to

any dependent loads. Stores may be processed through multiplexer (MUX) 106, which

has selectivity control, as illustrated in the sideways (horizontal) intersecting line from

the L1 STQ 102. (See FIG. 10, for example.)

[0027] According to the above embodiments, the present invention may include a first

level store queue 102 adapted to store in an n-entry buffer the last n stores in an

instruction window; and a second level circuit 104 adapted to accept and buffer non-

retired stores from the first level store queue 102. The first level store queue 102 further

includes an address matching circuit; and a store select circuit, where both circuits may

forward stores and store data to dependent loads. (See FIG. 10, for example.)

[0028] As one of ordinary skill in the art would realize, based at least on the teaching

provided herein, the first level store queue 102 may be a circular buffer with head and tail

pointers. (See FIG. 10, for example.)

3

[0029] In additional embodiments, the second level circuit, such as the speculative data cache 114, may include a memory dependence predictor (described further below) which may store in a non-tagged array one or more store-distances, wherein the store-distance may be the number of store queue entries between a load and a forwarding store. The second level circuit may also include an unresolved address buffer adapted to determine a program order condition. The program order condition may include if one or more non-issued load instructions are scheduled ahead of one or more associated store instructions. (See FIG. 11, for example.)

[0055] The system environment 800 may also include several processors, of which only two, processors 770, 780 are shown for clarity. Processors 770, 780 may each include a local memory channel hub (MCH) 772, 782 to connect with memory 702, 704. Processors 770, 780 may exchange data via a point-to-point interface 750 using point-to-point interface circuits 778, 788. Processors 770, 780 may each exchange data with a chipset 790 via individual point-to-point interfaces 752, 754 using point to point interface circuits 776, 794, 786, 798. Chipset 790 may also exchange data with a high-performance graphics circuit 738 via a high-performance graphics interface 792. Processors 770, 780 may include processor cores 774, 784, respectively.

4